Resistor-Temperature Conversion

Tesla cryogenic system local application

Wed, May 11, 1994

The TRTD local application supports needs of low temperature measurements for the Tesla superconducting RF laboratory in LAB-2 in the Fermilab village. Four signals are demultiplexed into 16, thermal emf's measured and used to get the resistance of the cold resistors, which are of two types, platinum and carbon. From the resistance, calibration tables are used with linear interpolation to get the temperature in degrees Kelvin. This note describes the methods used to do this.

Parameters layout

The available parameters for TRTD as seen via "Page E" are as follows:

```
E LOC APPL PARAMS 05/11/94 0838
NODE<056B>  NTRY< 5>  HELP<0576>
NAME=TRTD   CNTR=02AB
TITL"TEMPERATURE RESISTORS   "
SVAR=00046044
ENABLE  B<00C2> TRTD ENABLE
MPXDATA C<0120> T1MUX       V
V+,ETC  C<0200> HM1THV      V
R REF   C<01F0> RREFPT      OHM
MPXSEL  B<0128> RTD MPX SELECT 0
CURRDIR B<0120> RTD PT CURR DIR
EMF PER  <0258>
```

The usual enable Bit# argument is followed by MPXDATA, the base channel# of the four multiplexed signals. V+,ETC is the base channel# of the sixteen V^+, V^-, emf, resistance, and temperature values, altogether 5 sequential "arrays" of 16 channels. The R REF reference resistor channel# specifies the Pt reference resistor. The next channel# holds the Carbon reference resistor value. MPXSEL is the base Bit# of the digital byte used for selecting the four 1-of-4 multiplexer values. The program sequences this byte through values of 00,AA,55,FF on each four successive cycles. The CURRDIR parameter is the base Bit# of the two bits used for controlling the current direction through the Pt and C resistors, respectively. The EMF PER parameter is the number of cycles between measurements of the V^- and emf values, done by reversing the current direction for four cycles to collect these data.

Demultiplexing

Signals from the RTD system are very small, so they are amplified by four-input multiplexed amplifiers. Two-bit select codes are wired to each amplifier. The program operates all four two-bit select codes at once, as they share the same byte of digital output. Data is read from one selection each cycle, which for the LAB-2 facility is 10 Hz. In this way, all 16 channels of data that result are updated every 400 ms.

Thermal emf's

To measure the thermal emf's that result from the conductor leads that connect from room temperature to cold temperatures, it is necessary to reverse the current. Every so often, perhaps every few minutes, the current is reversed and the data collected for the negative direction of current flow. After taking four cycles to accomplish this, the current is returned to the usual positive direction. The formula for the emf is as follows:

$$emf = (V^+ + V^-)/2$$

These emf values are updated every time a new V^- is measured with the current direction negative.

Resistances

The resistors used are of two types, platinum and carbon. The single Pt resistor calibration table is fairly linear down to liquid nitrogen temperatures, so linear interpolation can be used with a set of 10 calibration points only. These Pt calibration points are part of the program source and are as follows:

```
kelvinPt[1]:= 14.0;   ohmsPt[1]:= 1.797;
kelvinPt[2]:= 20.0;   ohmsPt[2]:= 2.147;
kelvinPt[3]:= 30.0;   ohmsPt[3]:= 3.508;
kelvinPt[4]:= 40.0;   ohmsPt[4]:= 5.938;
kelvinPt[5]:= 50.0;   ohmsPt[5]:= 9.228;
kelvinPt[6]:= 70.0;   ohmsPt[6]:= 17.128;
kelvinPt[7]:= 100.0;  ohmsPt[7]:= 29.987;
kelvinPt[8]:= 150.0;  ohmsPt[8]:= 50.815;
kelvinPt[9]:= 200.0;  ohmsPt[9]:= 71.073;
kelvinPt[10]:= 300.0; ohmsPt[10]:= 110.452;
```

The formula for resistance is as follows:

$$R= ((V^{+} - emf)/V^{+}_{ref})*R_{ref}$$

Here V^{+}_{ref} is the voltage across the reference resistor (Pt or C) and R_{ref} the reference resistor value itself. The reference values are constants in the program. The values at the time of this writing are R_{ref} = 99.82 (Pt) and 1000.0 (C).

The carbon resistors calibration curve is very nonlinear, so more calibration points are used. Linear interpolation is done with 20 calibration points between the *log* of the temperature and the *log* of the resistance. In order to collect these calibration data for use by the TRTD program, the calibration data points were entered into Excel by editing the original full calibration data set text file. In the spreadsheet, it was easy to calculate an average of the 5 calibrated carbon resistors for use with those carbon resistors that have no measured calibration data. In addition, logs were calculated for the temperature and resistance calibration data points. These log values were output from Excel via a text file, which was then edited into an MPW assembly source file containing "DC.S" data directives. This source file was then assembled and the resulting data downloaded into the station that will run the TRTD local application. Upon initialization, the program requests this local data file and places the data into its calibration data arrays for use in converting the carbon resistances into temperatures. Part of the calibration data entered into Excel was this:

940214	T2	T3	T4	T5	T6	TAvg
293.477	107.54	106.33	107.70	108.29	101.33	106.24
77.828	127.02	125.42	126.61	127.68	123.28	126.00
56.171	139.35	137.44	138.79	139.94	135.31	138.17
46.159	148.57	146.50	147.92	149.07	144.34	147.28
…	…	…	…	…	…	…
4.686	823.95	794.60	815.01	812.69	789.51	807.15
4.254	960.07	923.74	948.93	945.27	918.79	939.36
3.849	1137.84	1092.06	1123.90	1118.58	1087.68	1112.01
2.597	2875.26	2729.90	2840.76	2798.62	2734.71	2795.85
2.083	5196.52	4879.04	5103.97	5036.64	4916.87	5026.61
1.813	7783.32	7266.99	7634.61	7521.43	7368.14	7514.90
1.527	13827.37	12689.70	13608.76	13288.20	13037.05	13290.22
1.257	28765.90	26017.65	28125.64	27273.40	26708.60	27378.24

The calculated logs of the calibration point data are as follows:

Log_{10}(temp)	Log_{10}(T2)	Log_{10}(T3)	Log_{10}(T4)	Log_{10}(T5)	Log_{10}(T6)	Log_{10}(TAvg)
2.4676	2.0316	2.0267	2.0322	2.0346	2.0057	2.0263
1.8911	2.1039	2.0984	2.1025	2.1061	2.0909	2.1004
1.7495	2.1441	2.1381	2.1424	2.1459	2.1313	2.1404
1.6643	2.1719	2.1658	2.1700	2.1734	2.1594	2.1681
…	…	…	…	…	…	…
0.6708	2.9159	2.9001	2.9112	2.9099	2.8974	2.9070
0.6288	2.9823	2.9655	2.9772	2.9756	2.9632	2.9728
0.5853	3.0561	3.0382	3.0507	3.0487	3.0365	3.0461
0.4145	3.4587	3.4361	3.4534	3.4469	3.4369	3.4465
0.3187	3.7157	3.6883	3.7079	3.7021	3.6917	3.7013
0.2584	3.8912	3.8614	3.8828	3.8763	3.8674	3.8759
0.1838	4.1407	4.1035	4.1338	4.1235	4.1152	4.1235
0.0993	4.4589	4.4153	4.4491	4.4357	4.4267	4.4374

These data were then edited as data declaration statements into an MPW assembly source file, assembled and downloaded as a data file of 32-bit floating point values called DATATRTD into the local station along with the LOOPTRTD local application. When the program is first initialized, it reads the data file for use during Carbon resistor temperature conversion.

Internal details

Four channels are read each cycle and converted as necessary. Of the 16 channels read, they are of the following “resistor types”:

Signal#	Mpx	Resistor type
1–4.	00	Pt, Pt, Pt, Pt
5–8.	01	CAvg, CAvg, CAvg, CAvg
9–12.	10	Pt, Pt, none, none
13–16.	11	C6, C2, C3, C4

The resistor type specifies which reference resistor is used as well as which calibration data to use to perform the temperature conversion.

During program operation, a data structure allocated at initialization keeps track of the context. Using Page E, one can find the location of this dynamically assigned data structure. Its structure at the time of this writing, expressed in the format produced by the memory dump page application, at a time when the signal values may not have been valid, is as follows:

```
M MEMORY DUMP     05/12/94 1056
056B:03D716 0420 0000 0000 0007   BlkSize, , ,BlkType
056B:03D71E 056B 0004 0000 0000   Node#, state#, wait, debug#
056B:03D726 0000 0085 0200 0210   currDir, dirWait, V+Chan, V-Chan
056B:03D72E 0220 0230 0240 1004   emfChan, resistChan, tempChan, dbI
056B:03D736 0000 00AA 0055 00FF   mpx, mpx, mpx, mpx
056B:03D73E 0000 0CC7 0000 7FFF   resRef(Pt), resRef(C)
056B:03D746 467F 8C00 44CC CB33   refConst(Pt), refConst(C)
056B:03D74E 7368 7480 03C3 0013   mpxChan(1–4)

056B:03D756 0101 0101 0707 0707   resTypes(1–16)
056B:03D75E 0101 0000 0602 0301
```

```
056B:03D766 0103 0102 0202 0102   deltaTimes(1–16)
056B:03D76E 0102 0102 0102 0202
056B:03D776 73EF 73CD 73AC 73C6   V+(1–16)
056B:03D77E 7374 7493 7398 73FD
056B:03D786 7393 7400 03C9 001C
056B:03D78E 73AC 73FE 73CA 73D4

056B:03D796 7387 73BB 73AC 73BD   V-(1–16)
056B:03D79E 738D 73E7 738B 73B9
056B:03D7A6 7374 7402 03C2 0021
056B:03D7AE 73A7 73F2 7249 738D
056B:03D7B6 7382 73D1 73B6 73D3   emf(1–16)
056B:03D7BE 7391 73F2 7386 73F9
056B:03D7C6 7380 73EE 03B5 001B
056B:03D7CE 73BC 73E8 72EB 7434

056B:03D7D6 016B 0604 FDB4 FEC8   resist(1–16)
056B:03D7DE 08BA E333 099A EF6B
056B:03D7E6 0140 023D 0000 0000
056B:03D7EE F71D 0AAB 3FA4 FD91
056B:03D7F6 0540 0CE9 0000 0000   kelvin(1–16)
056B:03D7FE 00E6 0042 00DC 0042
056B:03D806 0445 07A9 0000 0000
056B:03D80E 0042 00D5 006C 0000

056B:03D816 3FE6 0419 4009 6873   Pt calibration points
056B:03D81E 4060 8312 40BE 0419
056B:03D826 4113 A5E3 4189 0625
056B:03D82E 41EF E560 424B 428F
056B:03D836 428E 2560 42DC E76D
056B:03D83E 4160 0000 41A0 0000
056B:03D846 41F0 0000 4220 0000
056B:03D84E 4248 0000 428C 0000

056B:03D856 42C8 0000 4316 0000
056B:03D85E 4348 0000 4396 0000
056B:03D866 0000 0000 0000 0000   C calibration points
056B:03D86E 0000 0000 0000 0000
…           …    …    …    …
056B:03D906 4002 05BC 4006 A64C   Log10(Temp)
056B:03D90E 4009 38EF 400B 0069

056B:03D916 400D 7F63 4011 AEE6
056B:03D91E 4014 F41F 4019 B08A
056B:03D926 401E D5D0 4026 8241
056B:03D92E 402E 0C4A 4037 DD98
056B:03D936 403A 9E1B 403E DE01
056B:03D93E 4043 9724 405D 5B57
056B:03D946 406D CE07 4079 096C
056B:03D94E 4084 809D 408E AF4F

056B:03D956 4001 B574 4006 4C30   Log10(T2)
056B:03D95E 4008 D6A1 400A 9C78
056B:03D966 400D 119D 4011 35A8
056B:03D96E 4014 6DC6 4019 1EB8
056B:03D976 401E 339C 4025 C5D6
```

```
056B:03D97E 402D 374C 4036 E48F
056B:03D986 4039 9B3D 403D CAC1
056B:03D98E 4042 71DE 405B E910

056B:03D996 406C 0D1B 4077 212D
056B:03D99E 4083 4FDF 408D 4A23
056B:03D9A6 4002 0F91 4006 8F5C   Log10(T3)
056B:03D9AE 4009 1D15 400A E148
056B:03D9B6 400D 6388 4011 8ADB
056B:03D9BE 4014 CB29 4019 85F0
056B:03D9C6 401E A993 4026 4A8C
056B:03D9CE 402D CAC1 4037 923A

056B:03D9D6 403A 511A 403E 8A72
056B:03D9DE 4043 3EAB 405D 0481
056B:03D9E6 406D 4E3C 4078 7FCC
056B:03D9EE 4084 4817 408E 5F07
056B:03D9F6 4002 36E3 4006 CA58   Log10(T4)
056B:03D9FE 4009 566D 400B 18FC
056B:03DA06 400D 94AF 4011 B08A
056B:03DA0E 4014 F27C 4019 A1CB

056B:03DA16 401E B9F5 4026 511A
056B:03DA1E 402D C91D 4037 81D8
056B:03DA26 403A 3BCD 403E 703B
056B:03DA2E 4043 1DE7 405C 9A02
056B:03DA36 406C EF35 4078 154D
056B:03DA3E 4083 F3B6 408D F141
056B:03DA46 4000 5D64 4005 D14E   Log10(T5)
056B:03DA4E 4008 6738 400A 339C

056B:03DA56 400C AF4F 4010 D35B
056B:03DA5E 4014 13A9 4018 C2F8
056B:03DA66 401D DB23 4025 7C1C
056B:03DA6E 402C F5C3 4036 B368
056B:03DA76 4039 6F00 403D A512
056B:03DA7E 4042 5604 405B F62B
056B:03DA86 406C 44D0 4077 837B
056B:03DA8E 4083 AFB8 408D A787

056B:03DA96 4001 AEE6 4006 6CF4   Log10(T6)
056B:03DA9E 4008 FC50 400A C227
056B:03DAA6 400D 3F7D 4011 652C
056B:03DAAE 4014 A3D7 4019 59B4
056B:03DAB6 401E 7525 4026 147B
056B:03DABE 402D 8FC5 4037 4F0E
056B:03DAC6 403A 0C4A 403E 425B
056B:03DACE 4042 F34D 405C 9375

056B:03DAD6 406C E219 4078 0EBF
056B:03DADE 4083 F3B6 408D FF2E
056B:03DAE6 401D ED29 3FF2 0F91   Log10(TAvg)
056B:03DAEE 3FDF EF9E 3FD5 07C8
056B:03DAF6 3FC7 65FE 3FB3 CD36
056B:03DAFE 3FA6 D289 3F96 CBFB
056B:03DB06 3F88 4817 3F6D 5CFB
```

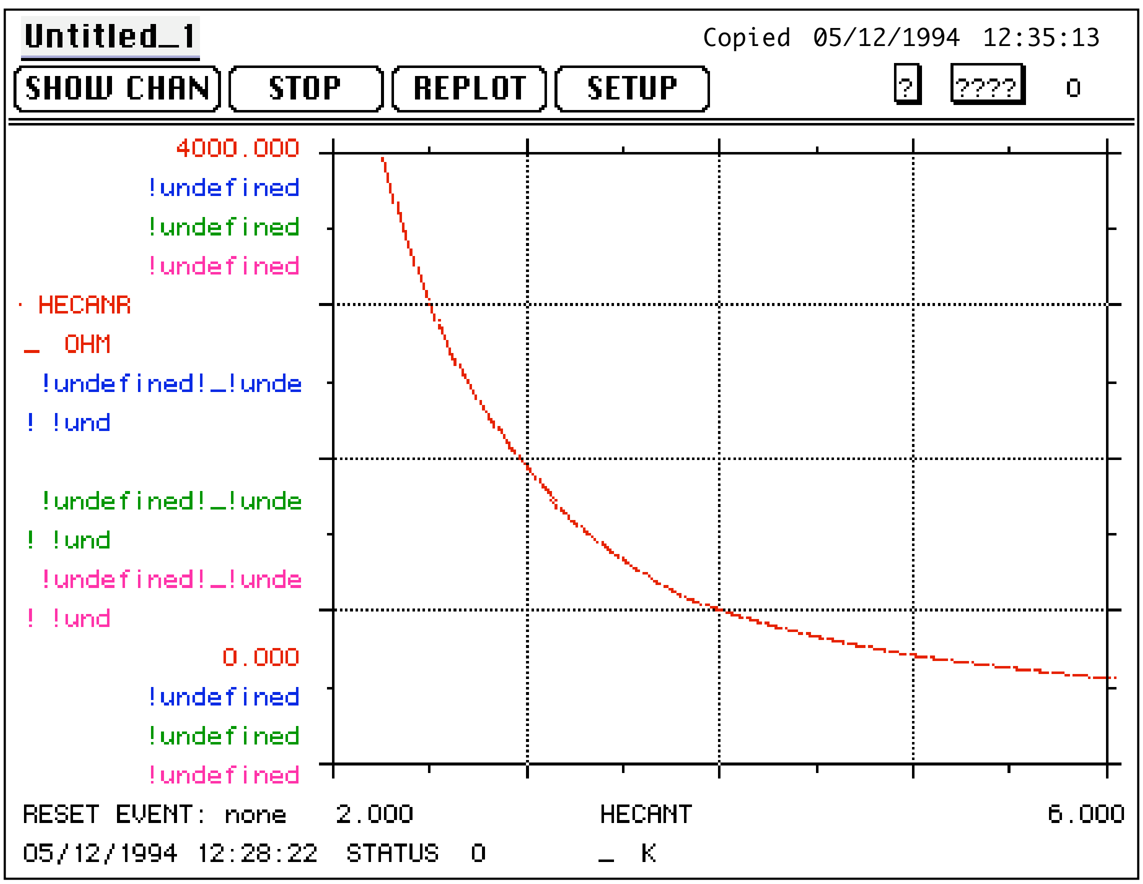
```
056B:03DB0E 3F51 4120 3F33 126F

056B:03DB16 3F2B B98C 3F20 F909
056B:03DB1E 3F15 D639 3ED4 3958
056B:03DB26 3EA3 2CA5 3E84 4D01
056B:03DB2E 3E3C 3611 3DCB 5DCC
```

The diagnostic `deltaTimes` byte array shows elapsed times per 10 Hz cycle in units of 0.5 ms. (A value of 02 means 1 ms.) Since the program operates on 4 signals per cycle, these times exhibit a periodicity of four cycles.

All floating point values are in IEEE standard 32-bit single precision.

Debug mode

A special debug mode is available for checking the ohms to temperature conversion for the Carbon resistors. To use it, find the above structure using Page E. Then set the debug#, which is initialized to zero, to the index value 1–16 of the channel whose conversion is to be checked. When this word is nonzero in this range, the program stops doing its usual work and concentrates on only the given channel index. It accepts the resistance value from the resistance channel, rather than computing a new one based upon the current demultiplexed data readings. This allows a value to be set into a resistance channel and the temperature channel observed resulting from this conversion. One can make a calibration plot of temperature versus resistance in this way using, for example, the Parameter Page on the Macintosh written by Bob Peters. Here is an example of making such a calibration curve for the T6 signal:



Logarithms, etc

Because of the extremely nonlinear characteristic of the resistance of the Carbon resistors relative to temperature, it was necessary to work with logarithms of these parameters. With linear interpolation on the *logarithms*, reasonably accurate results can be obtained. But the IRMs use a 68040 cpu, which does not have support for logarithms and powers on-chip, although it does have support for the basic add, subtract, multiply and divide. So a routine was needed to compute logarithms and powers. In a book called "Approximations for Digital Computers" by Cecil Hastings, Jr., Princeton University Press, 1955, can be found suitable formulae for these and other functions to several degrees of accuracy. The formula for $Log_{10}(x)$ that was used in TRTD has an error function < 0.000002 over the range $1 \le x \le 10$. Its form is as follows:

$$Log_{10}(x) = 0.5 + c_1*f + c_3*f^3 + c_5*f^5 + c_7*f^7,$$

where $f = (x - k)/(x + k)$, and $k = \sqrt{10} = 3.162278$. The set of coefficients is:

$c_1 = 0.86855434$; $c_3 = 0.29115068$; $c_5 = 0.15361371$; $c_7 = 0.21139497$.

The formula for 10^x that was used has an error function < 0.000003 over the range $0 \le x \le 1$. Its form is as follows:

$$10^x = (1 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5)^2$$

The set of coefficients is:

$a_1=1.15138424$; $a_2=0.66130851$; $a_3=0.26130650$; $a_4=0.05890681$; $a_5=0.02936622$;

The TRTD program has about 600 lines of Pascal and compiles into about 4K bytes.